Dynamic visualization of high-dimensional data via low-dimension projections and sectioning across 2D and 3D display devices

Candidature confirmation document for the degree of

Doctor of Philosophy

by

Nicholas S Spyrison

B.Sc. Statistics, Iowa State University

Department of Information Technology

Monash University

Australia

February 2019

Contents

Declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma in any university or equivalent institution, and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Nicholas S Spyrison

Abstract

Visualizing data space is crucial to exploratory data analysis yet doing so quickly becomes difficult as the dimensionality of the data increases. Traditionally, static, low-dimensional linear embeddings are used. Observing one such embedding often misses a significant amount of variation, and hence, information held within the data. Touring is a method that animates many projections as the orientation in data space is varied. This maintains transparency to the original variables, while preserving information in the data.

There are various ways to generate tour paths and many geometrics to view each path. Once an interesting feature has been identified more information can be gleaned with finer control to explore the local structure, in particular through User Controlled Steering (UCS).

This research has implemented UCS in 2D embeddings and will further compare 2D UCS with alternatives, bring UCS to 3D embeddings in virtual reality, and explore the efficacy of doing so.

Research problem

Data and models are typically high-dimensional, with many variables or parameters. Developing new methods to visualize high dimensions have been a pursuit of statisticians, computer scientists and visualization researchers for decades. As technology evolves examining, extending, and assessing current techniques, in new environments, for new data challenges, is an important endeavor.

This thesis focuses on tour methods for visualizing high-dimensional data. Tours are a family of algorithms for generating paths on the space of low-dimensional ($d = 1, 2, 3, ..., p$) projections of high-dimensional (p) space. The resulting projection of the data (or model) is displayed using low-dimensional techniques such as histograms, dot plots, scatterplots, or parallel coordinate plots, and the path generates a movie or animation to shows many low-dimensional projections. The method can be applied with other techniques such as machine learning techniques like discriminant analysis, neural networks, support vector machines, to open the black box, and complements dimension reduction techniques like principal component analysis (PCA), multidimensional scaling (MDS) on nonlinear embeddings (e.g. tSNE).

The primary research question is:

Does allowing the user to steer the tour, facilitate the exploration and understanding of the sensitivity of structure to the original variables or parameters?

This question is reduced to four research objectives:

RO #A **Can user controlled steering (UCS) be utilized in an environment providing non-dynamic animation?** (Chapter 2) This is previous experimental design that extend dynamic projection functionality currently available in the programming language R. It implements UCS steering and extends the animation paradigms that can be used for touring. These paradigms fall short of dynamic user interaction.

RO #B **What benefits does UCS provide over popular alternatives?** (Chapter 3) Tentative planned future work of a case study comparing USC with commonly used static alternatives, such as principal component analysis (PCA), linear discriminant analysis (LDA), multi-dimensional scaling (MDS), and t-distributed nearest neighbor embeddings (tSNE).

RO #C **How do we extend UCS to 3D?** (Chapter 4) This future experimental design will scaling UCS to 3 dimensions (novel contribution) and extend upon available functionally in the game engine Unity for compatible front ends across display devices.

RO #D **Does UCS with 3D displays provide perception benefits over 2D displays?** (Chapter 5) This future empirical study explores the efficacy of bringing USC into 3D as compared across various display devices.

0.1 Methodology

As outlined above, RO #A & C uses *experimental design* in R and R & C# Unity respectively. RO #B is a *case study* comparison between UCS and alternatives. An *empirical research* explores the efficacy of UCS across display type in RO #D.

Chapter 1

Literature review

1.1 Dynamic projections of multivariate data (Touring)

1.1.1 Overview

In univariate data sets histograms, or smoothed density curves are employed to visualize data. In bivariate data $x - y$ scatterplots and contour plots (2D density) can be employed. In three dimensions the two most common techniques are 3D scatterplot[1] or 2D scatterplot with the 3rd variable as an aesthetic (such as, color, size, height, *etc.*). Mapping variables to data point aesthetics can afford another dimension or 2, but this is not a sustainable solution.

How do we visualize data for even modest values of p numeric dimension (say 6 or 12)? It's far too common that visualizing in data-space is dropped altogether in favor of modeling parameter-space, model-space, or worse: long tables of statistics without visuals (Wickham, Cook, and Hofmann, 2015). Yet, there are inherent risks inherent in relying too heavily on summary statistics alone (Anscombe, 1973; Matejka and Fitzmaurice, 2017). So why do we move away from visualizing in data-space? Scalability, in a word. Humans are not equipped to visualize above 3 spatial dimensions. This is where dimensionality reduction comes in. We will focus on a group of linear orthogonal projection techniques know as

[1] Graphs depicting 3 dimensions are typically printed on paper, or rendered on a 2D monitor, they are intrinsically 2D images of monocular 3D spaces, sometimes referred to as 2.5D, but more frequently referred to as 3D, more on this in section 1.2.1.

touring. For a broader view of dimensionality reduction techniques, see Grinstein, Trutschl, and Cvek (2002), Carreira-Perpinán (1997), or Heer, Bostock, and Ogievetsky (2010). We choose touring, for a couple of salient features: use of linear projections maintaining transparency back to the original variable space (which non-linear projections lose) and keeps all components and their information in tack (which static linear projections lose). By employing the breadth of tours, we can preserve the visualization of data-space, and with it, the intrinsic understanding of structure and distribution of data that is more succinct or beyond the reach of statistic values alone.

Touring is a linear dimensionality reduction technique that orthogonally projects p-space down to $d(\leq p)$ dimensions. Many such projections are interpolated, each making small rotations in p-space. These frames are then viewed in order as an animation of the lower dimensional embedding changing as the original variable space is manipulated. Shadow puppets offer a useful analogy to aid in conceptualizing touring. Imagine a fixed light source facing a wall. When a hand or puppet is introduced the 3-dimensional object projects a 2D shadow onto the wall. This is a physical representation of a simple projection, that from $p = 3$ down to $d = 2$. If the object rotates then the shadow correspondingly changes. Observers watching only the shadow are functionally watching a $d = 2$ tour as the $p = 3$ object is manipulated. In some views more information is hidden than in others but watching an animation of the shadow changing gives a more robust understanding than looking at any one still frame. More complex structures generally require more time to comprehend the nature of the geometry. These features hold true in touring as well.

1.1.2 Terminology

Terminology varies across articles. In my work, I use the following:

- n, number of observations in the data.
- p, number of numeric variables, the dimensionality of data space.
- d, dimensionality of projection space.
- $\mathbf{X}_{[n,\ p]}$, a data matrix in variable-space, $\mathbf{X} \in \mathbb{R}^p$. Typically centered, scaled, and optionally sphered.

- $B_{[p,\ d]}$, orthonormal basis set (d othogonal linear combinations of length p, each with a norm of 1), defining the orientation of projection from $p-$ to $d-$space.
- $\mathbf{Y}_{[n,\ d]}$, projected data matrix in projection-space, $\mathbf{Y} \in \mathbb{R}^d$.
- For projections down to 1- and 2D, it's common to display each variables contribution and direction on its own axis (1D) or relative to a unit circle (2D), this is referred to as basis axes or sometimes the reference frame.
- Geometric objects are referred to in generalized dimensions; the use of plane isn't necessarily a 2D surface, but a hyper-plane in the arbitrary dimensions of the projection space.

1.1.3 History

Touring was first introduced by Asimov (1985) with his purposed *grand tour* at the Stanford Linear Accelerator, Stanford University. In which, Asimov suggested three types of grand tours: torus, at-random, and random-walk. The original application of touring was on high energy physics on the PRIM-9 system.

Before choosing projection paths randomly, an exhaustive search of $p-$space of was suggested by McDonald (1982), also at the Stanford Linear Accelerator. This was later coined as *little tour*.

Friedman and Tukey (1974) and later Huber (1985) purposed projection pursuit (also referred to as PP). Projection pursuit involves identifying "interesting" projection, remove a single component of the data, and then iterates in this newly embedded subspace. Within each subspace the projection seeks for a local extremum via hill climbing algorithm on an objective function. This formed the basis for *guided tours* suggested by Hurley and Buja (1990).

The grand and little tour have no input from the user aside from the starting basis. Guided tours allow for an index to be selected, but the bulk of touring development since has largely been around dynamic display, user interaction, geometric representation, and application. The extent to which will be expounded on in the following sections.

1.1.4 Path generation

A fundamental aspect of touring is the path of rotation. Of which there are four primary distinctions (Buja et al., 2005): random choice, data driven, precomputed choice, data driven, and manual control.

- Random choice, *grand tour*, constrained random walks p-space. Paths are constrained for changes in direction small enough to maintain continuity and aid in user comprehension
  - torus-surface (Asimov, 1985)
  - at-random (Asimov, 1985)
  - random-walk (Asimov, 1985)
  - *local tour* (Wickham et al., 2011), a sort of grand tour on a leash, such that it goes to a nearby random projection before returning to the original position and iterating to a new nearby projection.
- data driven, *guided tour*, optimizing some objective function/index via projection pursuit (Hurley and Buja, 1990), analogous to gradient descent. including the following indexes:
  - holes (Cook, Buja, and Cabrera, 1993) - iterates projections that add more white space to the center of the projection.
  - cmass (Cook, Buja, and Cabrera, 1993) - find the projection with the most density or mass in the center.
  - lda (Lee et al., 2005) - linear discriminant analysis, seeks a projection where 2 or more classes are most separated.
  - pda (Lee and Cook, 2010) - penalized discriminant analysis for use in highly correlated variables when classification is needed.
  - convex (Laa and Cook, 2019) - the ratio of area of convex and alpha hulls.
  - skinny (Laa and Cook, 2019) - the ratio of of the perimeter distance to the area of the alpha hull.
  - stringy (Laa and Cook, 2019) - based on the minimum spanning tree (MST), the diameter of the MST over the length of the MST.

  - dcor2D (Grimm, 2017; Laa and Cook, 2019) - distance correlation that finds linear and non-linear dependencies between variables.
  - splines2D (Grimm, 2017; Laa and Cook, 2019) - measure of non-linear dependence by fitting spline models.
  - other user-defined objective function can be implemented with the *tourr* package Wickham et al. (2011).

- Precomputed choice, *planned tour*, in which the path has already been generated or defined.

  - *little tour* (McDonald, 1982), where every permutation of variables is stepped through in order, analogous to a brute-force or exhaustive search.
  - a saved path of any other tour, typically an array of basis targets to interpolate between as produced by the R function `tourr::save_history()`.

- Manual control, *manual tour*, a constrained rotation on selected manipulation variable and magnitude (Cook and Buja, 1997). Typically used to explore the local area after identifying an interesting feature from another tour.
- *dependence tour*, combination of n independent 1D tours. A vector describes the axis each variable will be displayed on. *ie* $c(1,1,2,2)$ is a 4- to 2D tour with the first 2 variables on the first axis, and the remaining on the second.

  - *correlation tour* (Buja, Hurley, and McDonald, 1987), a special case of the dependence tour, analogous to canonical correlation analysis.

1.1.5 Path evaluation

Consider $d = 2$, then each projection is called a 2-frame (each spanning a 2-plane). Mathematically we call the set of all possible unoriented 2-frames in p-space a Grassmannian, $\mathbf{Gr}(2,\ p)$. Asimov (1985) pointed out that the unique 2-frames of the grand tour approaches $\mathbf{Gr}(2,\ p)$ as time goes to infinity. We could then define the *density* of a tour as the fraction of the Grassmannian explored. Ideally a grand tour will be dense, but the time taken to become dense vastly increases as variable space increase dimensionality. We could then also talk about the *rapidity* of a tour as how quickly a tour encompasses the Grassmannian. Due to the random selection of a grand tour it will end up visiting homomorphisms of previous 2-frames, sub-optimal rapidity.

The little tour introduced in McDonald (1982), on the other hand is necessarily both dense and rapid, performing essentially an exhaustive search on the Grassmannian. However, this path uninteresting and with long periods of similar projections strung together. There was a need to find interesting projections quicker.

Guided tours (Hurley and Buja (1990)) optimize an objective function generating path will be relatively small subset of the Grassmannian, density and rapidity become poor measures, while interesting projections are quickly identified. Recently, Laa and Cook (2019), compares projection pursuit indices with the metrics: smoothness, squintability, flexibility, rotation invariance and speed. See the original work to see how the indices perform.

1.1.6 Geometric display by dimensionality

Up to this point we have been talking about 2D scatterplots, which offer the first and a simple case for viewing lower-dimensional embeddings of p-space. However, other geometrics (or geoms) offer perfectly valid projections as well.

- 1D geoms
  - 1D densities: such as histogram, average shifted histograms (Scott, 1985), and kernel density (Scott, 1995).
  - image (pixel): (Wegman, Poston, and Solka, 2001)
  - time series: where multivariate values are independently lagged to view peak and trough alignment. Currently no package implementation, but use case is discussed in (Cook and Buja, 1997).
- 2D geoms
  - 2D density (available on GitHub at `https://github.com/nspyrison/tourr`)
  - $x - y$ scatterplot

- 3D geoms - these geoms do not perform projections in 3 dimensions, but rather are $d = 2$ projections that utilize the manipulation dimension to give depth perception cues.
  - Anaglyphs, sometimes called stereo, where (typically) red images are positioned for the left channel and cyan for the right, when viewed with corresponding filter glasses give the depth perception of the image.
  - Depth, which gives depth cues in aesthetic mappings, most commonly size and/or color of data points.
- d-dimensional geoms
  - Andrews curves (Andrews, 1972), smoothed variant of parallel coordinate plots, discussed below.
  - Chernoff faces (Chernoff, 1973), variables linked to size of facial features for rapid cursory like-ness comparison of observations.
  - Parallel coordinate plots (Ocagne, 1885), where any number of variables are plotted in parallel with observations linked to their corresponding variable value by polylines.
  - Scatterplot matrix (Becker and Cleveland, 1987), showing a triangle matrix of bivariate scatterplots with 1D density on the diagonal.
  - Radial glyphs, radial variants of parallel coordinates including radar, spider, and star glyphs (Siegel et al., 1972).

1.1.7 Tour software implementations

Below is a non-exhaustive list of software implementing touring in some degree, ordered by descending year:

- spinifex github.com/nspyrison/spinifex – R package, all platforms.
- tourr (Wickham et al., 2011) – R package, all platforms.
- CyrstalVision (Wegman, 2003) – for Windows.
- GGobi (Swayne et al., 2003) – for Linux and Windows.
- DAVIS (Huh and Song, 2002) – Java based, with GUI.

- ORCA (Sutherland et al., 2000) – Extensible toolkit build in Java.
- VRGobi (Nelson, Cook, and Cruz-Neira, 1998) – for use with the C2, tours in stereoscopic 3D displays.
- ExplorN (Carr, Wegman, and Luo, 1996) – for SGI Unix.
- XGobi (Swayne, Cook, and Buja, 1991) – for Linux, Unix, and Windows (via emulation).
- XLispStat (Tierney, 1990) – for Unix and Windows.
- Explor4 (Carr and Nicholson, 1988) – Four-dimensional data using stereo-ray glyphs.
- Prim-9 (Asimov, 1985; Fisherkeller, Friedman, and Tukey, 1974) – on an internal operating system.

Support and maintenance of such implementations give them a particularly short life span, while conceptual abstraction and technically heavier implementations have hampered user growth. There have been notable efforts to diminish the barriers to entry and make touring more approachable as a data exploration tool (Huh and Song, 2002; Swayne et al., 2003; Wegman, 2003; Wickham et al., 2011; Huang, Cook, and Wickham, 2012).

1.1.8 Going further

Most previous implementation of dynamic projections are failing to compile. I have extended the graphic options for touring dynamic projections and reimplemented UCS on a compatible platform for more sustainable support (RO #A). A comparative study outlining the benefits of UCS vs alternatives is also absent from the literature (RO #B).

1.2 Multivariate data visualization in 3D

As this research pertains to numeric multivariate data, we'll stick to this scope in the literature review. For wider overview of 3D data visualization see chapter 2 of Marriott et al. (2018).

1.2.1 Terminology

- 2D - representation of data in 2 dimensions, without use of depth perception cues and minimal aesthetic mapping (color, size, height, *etc*) to data points.
- 2.5D - Following the definition given in Ware (2000) : visualizations that are essentially 2D, but select depth cues are used to provide some suggestion of 3D. However, the term 2.5D is commonly used for several meanings *due to the ambiguity use of 2.5D, we err on the side stating 3D with descriptions of depth cues used*.
- 3D - visualizations of 3 dimensions with a liberal use of depth cues unless otherwise qualified.
- Depth perception cues - an indication that tips off depth to an observer, including:
  - linear perspective - the property of parallel lines converging on a vanishing point.
  - aerial perspective - objects that are far away have lower contrast and color saturation due to light scattering in the atmosphere.
  - occulation (or interposition) - where closer objects partially block the view of further objects.
  - motion perspective/parallax - closer objects, move across the field of view faster than further objects.
  - accommodation - the change of focal length due to change in the shape of the eye. Effective for distances of less than 2 meters.
  - binocular stereopsis/disparity - the use of 2 images of slightly varied angles from the horizontal distance of the eyes. The disparity for distant objects is small, but it significant for nearby objects.
  - binocular convergence - The ocular-motor cue due to stereopsis focusing on the same objects. Convergence is effective for distances up to 10 meters.
- Virtual reality (VR) - computer generated display of virtual spaces in place of physical vision.
- Augmented reality (AR) - computer generated display of information over laid on a physical space.

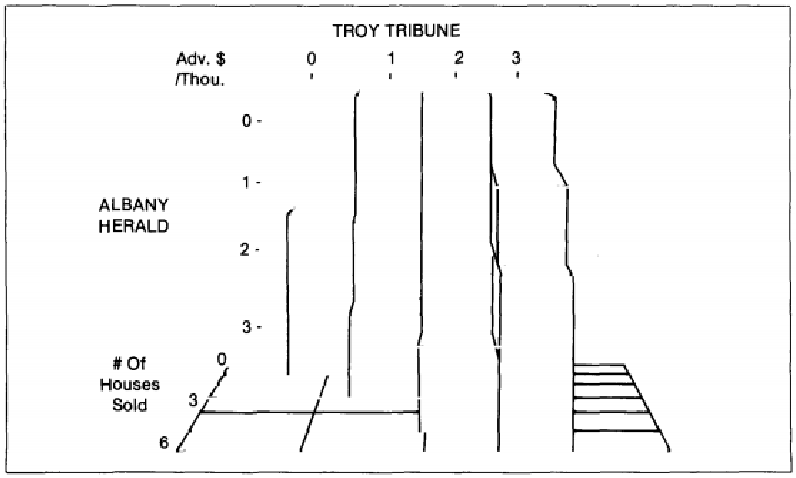


Figure 1.1: *Screen capture of "Figure 7. 3-D Block Model" from Lee, MacLachlan, and Wallace (1986).*

- Mixed reality (MR) - any degree of virtual or augmented reality.
- Scatterplot matrices (SPLOM) - matrix display of pair-wise 2D scatterplots.

1.2.2 A rocky start

Scientific visualization has readily adopted mixed realities as a large amount of the science exist in 3 spatial dimensions, lending itself well to virtual immersion. Data visualization, on the other hand, has been slow to utilize graphics above 2.5D, (and haptic interaction) primarily due to the mixed results of over-hyped of 3D visuals from the 1980's and 90's (Munzner, 2014). However, since then there have been several promising studies suggesting that it's time for data visualization to revisit and adopt 3D visuals for specific combinations of visuals and depth cues.

1.2.3 3D rotated projections vs 3 2D orthogonal projections

3D shapes can be represented by 3 orthogonal 2D views, or rather 3 pairwise projections. When 3D representations are used with binocular cues, they are found to have more accurate perception than 2D counterparts (Lee, MacLachlan, and Wallace, 1986, depicted in figure 1.1).

Between 3D and split view 2D of 3-dimensional economics data Wickens, Merwin, and Lin (1994), depicted in figure 1.2, asked participants integrative questions, finding that

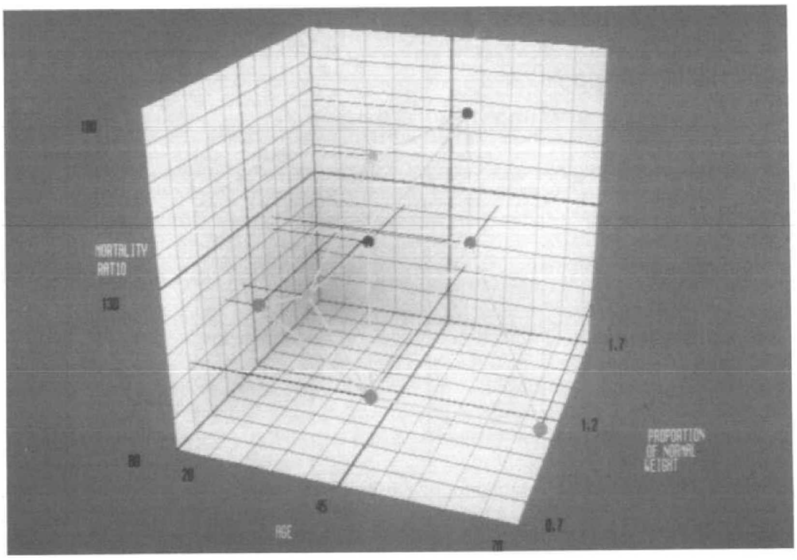


Figure 1.2: *Screen capture of "Figure 5. Example of a mesh display" from Wickens, Merwin, and Lin (1994).*

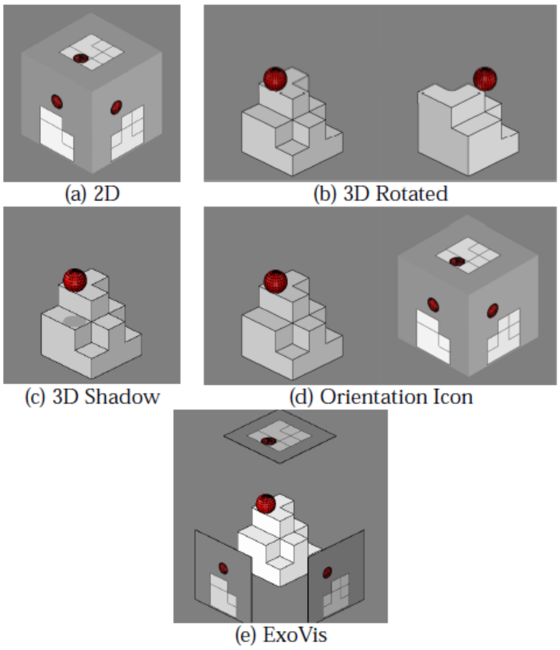
participants were faster to answer when questions involved three dimensions, while performance was similar when questions involved fewer dimensions.

Using 3D rotated projection gives accurate perception of a ball relative to complex box shapes, while combinations of 2D and 3D give the most precise orientation and positioning information (Tory et al., 2006, depicted in figure 1.3).

Sedlmair, Munzner, and Tory (2013), depicted in figure 1.4, tasked users with cluster separation across 2D scatterplot, 2D scatterplot matrices (SPLOMs) and interactive 3D scatterplots as viewed in monocular 3D from a standard monitor. They conclude that interactive 3D scatterplots perform worse for class separation. This result is surprisingly as the extra dimension theoretically allows for clustering structure to be seen and explored more clearly.

1.2.4 Comparing 3D and 2D embeddings of multivariate data

Nelson, Cook, and Cruz-Neira (1998), depicted in figure 1.5, had $n = 15$ participants perform brushing and touring tasks (identification of clusters, structure, and data dimensionality) in 3D with head-tracked binocular VR. 3D proved to have substantial advantage for cluster identification and some advantage in in identifying shape. Brushing did take longer in VR, perhaps due to the lower familiarity of manipulating 3D spaces.



(a) 2D

(b) 3D Rotated

(c) 3D Shadow

(d) Orientation Icon

(e) ExoVis

Figure 1.3: *Screen capture from Tory et al. (2006): "Fig. 1 (a) 2D, (b) 3D Rotated, (c) 3D Shadow, (d) Orientation Icon, and (e) ExoVis displays used in Experiment 1 (position estimation). Participants estimated the height of the ball relative to the block shape. In this example, the ball is at height 1.5 diameters above the block shape."*

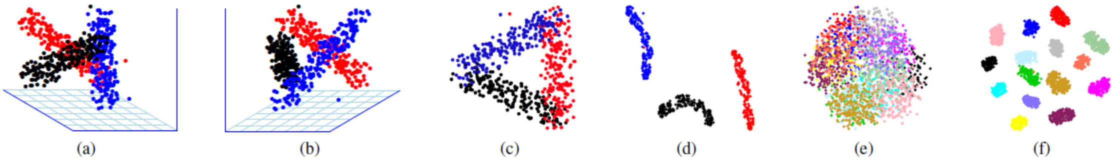


Figure 1.4: *Screen capture of "Figure 5. Example of a mesh display" from Sedlmair, Munzner, and Tory (2013): "Fig. 5. (a)-(d): Screenshots of the entangled dataset* `entangled1-3d-3cl-separate` *designed to show the most possible benefits for i3D. (a),(b) two viewpoints of the same i3D PCA scatterplot. An accompanying video shows the full 3D rotation. (c) 2D PCA projection. (d) t-SNE untangles this class structure in 2D. (e)-(f): 2D scatterplots of the reduced* `entangled2-15d-adjacent` *dataset which we designed to have a ground truth entangled class structure in 15D. (e) Glimmer MDS cannot untangle the classes, neither can PCA and robPCA (see supplemental material). (f) t-SNE nicely untangles and separates the ground truth classes in 2D."*

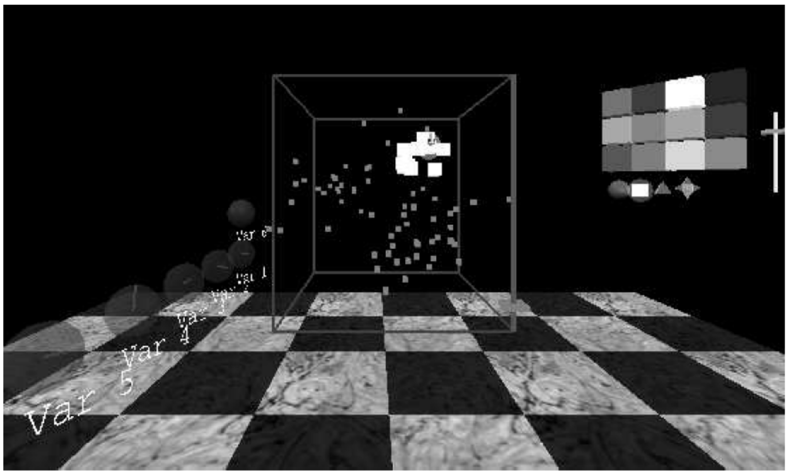


Figure 1.5: *Screen capture from Nelson, Cook, and Cruz-Neira (1998): "Figure 4: This is a picture of a 3-D room, running VRGobi. Data is plotted in the center, with painting tools to the right and variable spheres to the left. In the viewing box the data can be seen to contain three clusters, and one is being brushed."*

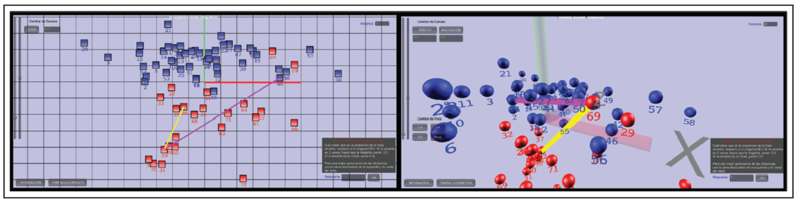


Figure 1.6: *Screen capture from Gracia et al. (2016): "Figure 5. Distance perception test. Left-hand image: 2D version. Here, the yellow line could be perceived as roughly twice the length of the magenta line, thus the value to be introduced should be approximately 2.0. Right-hand image: 3D version. Here, the inclusion of an extra dimension could provide new information about the relation, in terms of distances, between both lines."*

Another study, Gracia et al. (2016), depicted in figure 1.6, performed dimensionality reduction down to 2- and 3D scatterplots, both displayed in monocular 3D on a standard monitor. Users were found to more accurately compare distances between points and identify outliers on 3D scatterplots. However, both tasks were performed slower with use of the 3D scatterplots and statistical significance was not reported.

Wagner Filho et al. (2018), depicted in figure 1.7, performed an $n = 30$ empirical study of PCA embedded projections, and perception error across 4 tasks and 3 display types: 2D, 3D, and immersive. Overall task error was less in 3D and immersive relative to 2D. According to user Likert-scale 2D is slightly easier to navigate and slightly more comfortable. Conversely, 3D and immersive are slightly easier to interact and moderately easier to find information.

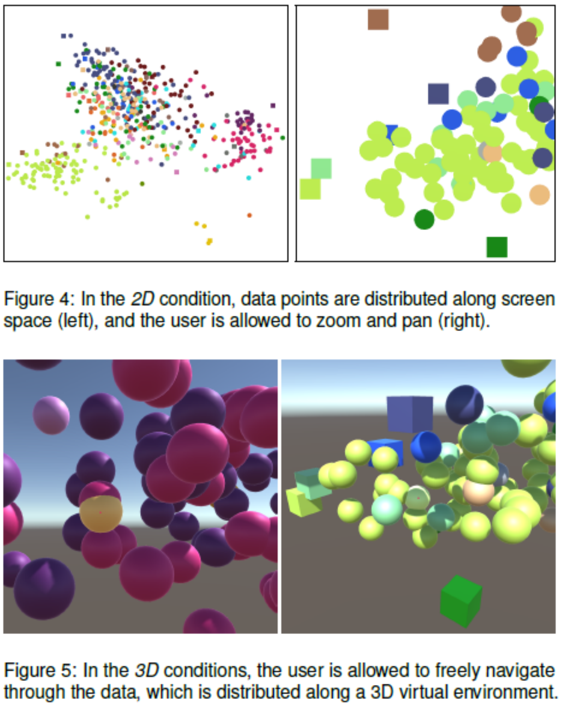


Figure 4: In the *2D* condition, data points are distributed along screen space (left), and the user is allowed to zoom and pan (right).

Figure 5: In the *3D* conditions, the user is allowed to freely navigate through the data, which is distributed along a 3D virtual environment.

Figure 1.7: *Screen capture from Wagner Filho et al. (2018), original captions contained in capture.*

1.2.5 Going further

When comparing between 3D and 2D orthogonal views studies in general show that perception accuracy is better in 3D, though manipulation speed is generally slower, confounded by users being less familiar with manipulating 3D space (Lee, MacLachlan, and Wallace, 1986; Wickens, Merwin, and Lin, 1994; Tory et al., 2006; counter example Sedlmair, Munzner, and Tory, 2013). Similar results have been shown in static, 3D embedded spaces (Gracia et al., 2016; Wagner Filho et al., 2018) and in dynamic 2D embedded spaces depicted in immersive 3D (Nelson, Cook, and Cruz-Neira, 1998). After more than 2 decades of hardware advancement, the quality, support, and prevalence of VR technology is better and more pervasive than ever. It's time to view dynamic 3D projections in immersive spaces and quantify the benefits (RO #B & C).

Chapter 2

Using animation to explore sensitivity of structure in a low-dimensional projection of high-dimensional data with user controlled steering

The content contained in this chapter is work done in the last year of my research and currently formatted as a paper to be submitted to the R Journal.

2.1 Abstract

The tour algorithm, and its various versions provide a systematic approach to viewing low-dimensional projections of high-dimensional data. It is particularly useful for understanding multivariate data, and useful in association with techniques for dimension reduction, supervised and unsupervised classification. The *R* package *tourr* provides many methods for conducting tours on multivariate data. This paper discusses an extension package which adds support for the manual tour, called *spinifex*. It is particularly usefully for exploring the sensitivity of structure discovered in a projection by a guided tour, to the contribution of a variable. *Spinifex* utilizes the animation packages *plotly* and *gganimation* to allow users to rotate the selected variable into and out of a chosen projection.

Keywords: grand tour, projection pursuit, manual tour, high dimensional data, multivariate data, data visualization, statistical graphics, data science, data mining.

2.2 Introduction

A tour is a multivariate data analysis technique in which is a sequence of linear (orthogonal) projections into a lower subspace in which $p-$space is rotated across time. Each frame of the sequence corresponds to a small change in the projection for a smooth transition to persevere the object continuity.

Multivariate data analysis can be broken into 2 groups: linear and non-linear transformations. Like PCA and LDA, touring uses linear dimension reduction that maintain transparency back to the original variable-space. PCA and LDA are typically represented with single static projection as a 2- or 3D scatterplot, inherently losing the variation held with the high components, whereas touring keep the information in tack by showing the other components across time. Non-linear transformations such as tSNE (t-distributed stochastic nearest neighbor embeddings), MDS (multi-dimension scaling), and LLE (local linear embedding) distort the parameter-space which lacks transparency back to the original parameter-space. They show more extreme separation in embeddings, but the variable opacity can be a non-starter for many uses.

There are many ways that a tour path can be generated, we will focus on one, the manual tour. The manual tour was described in Cook and Buja (1997) and allows a user to rotate a variable into and out of a 2D projection of high-dimensional space. This will be called user-controlled steering (UCS). The primary purpose is to determine the sensitivity of structure visible in a projection to the contributions of a variable. Manual touring can also be useful for exploring the local structure once a feature of interest has been identified, for example, by a guided tour (Cook et al., 1995). The algorithm for a manual tour allows rotations in horizontal, vertical, oblique, angular and radial directions. Rotation in a radial direction, would pull a variable into and out of the projection, which allows for examining the sensitivity of structure in the projection to the contribution of this variable. This type of manual rotation is the focus of this paper.

A manual tour relies on user input, and thus has been difficult to program in R. Ideally, the mouse movements of the user are captured, and passed to the computations, driving the rotation interactively. However, this type of interactivity is not simple in R. This has been the reason that the algorithm was not incorporated into the *tourr* package. Spinifex utilizes two new animation packages, *plotly* (Sievert, 2018) and *gganimate* (Pedersen and Robinson, 2019), to display manual tours or other saved tours. From a given projection, the user can choose which variable to control, and the animation sequence is generated to remove the variable from the projection, and then extend its contribution to be the sole variable in one direction. This allows the viewer to assess the change in structure induced in the projection by the variable's contribution.

The paper is organized as follows. Section 2.3 explains the algorithm using a toy dataset. Section 2.5 illustrates how this can be used for sensitivity analysis. The last section, **??** summarizes the work and discusses future research.

2.3 Algorithm

Creating a manual tour animation requires these steps:

1. Provided with a 2D projection, choose a variable to explore. This is called the "manip" variable.
2. Create a 3D manipulation space, where the manip variable has full contribution.
3. Generate a rotation sequence which zero's the norm of the coefficient and increases it to 1.

These steps are described in more detail below.

2.3.1 Notation

This section describes the notation used in the algorithm description. The data to be displayed is an $n \times p$ numeric matrix.

$$\mathbf{X}_{[n,\, p]} = \begin{bmatrix} X_{1,\,1} & \dots & X_{1,\,p} \\ X_{2,\,1} & \dots & X_{2,\,p} \\ \vdots & \ddots & \vdots \\ X_{n,\,1} & \dots & X_{n,\,p} \end{bmatrix}$$

An orthonormal d-dimensional basis set is describing the projection from $p-$ to $d-$ space

$$\mathbf{B}_{[p,\, d]} = \begin{bmatrix} B_{1,\,1} & \dots & B_{1,\,d} \\ B_{2,\,1} & \dots & B_{2,\,d} \\ \vdots & \ddots & \vdots \\ B_{p,\,1} & \dots & B_{p,\,d} \end{bmatrix}$$

The algorithm is primarily operating on the projection basis and utilizes the data only when making a display.

2.3.2 Toy data set

The flea data from the R package *tourr* (Wickham et al., 2011), is used to illustrate the algorithm. The data, originally from Lubischew (1962), contains 74 observations across 6 variables, which physical measurements of the insects. Each observation belonging to one of three species.

A guided tour on the flea data is conducted by optimizing on the `holes` index (Cook, Swayne, and Buja, 2007). In a guided tour the data the projection sequence is shown by optimizing an index of interest. The holes index is maximized by when the projected data has a lack of observations in the center. Figure 2.1, shows an optimal projection

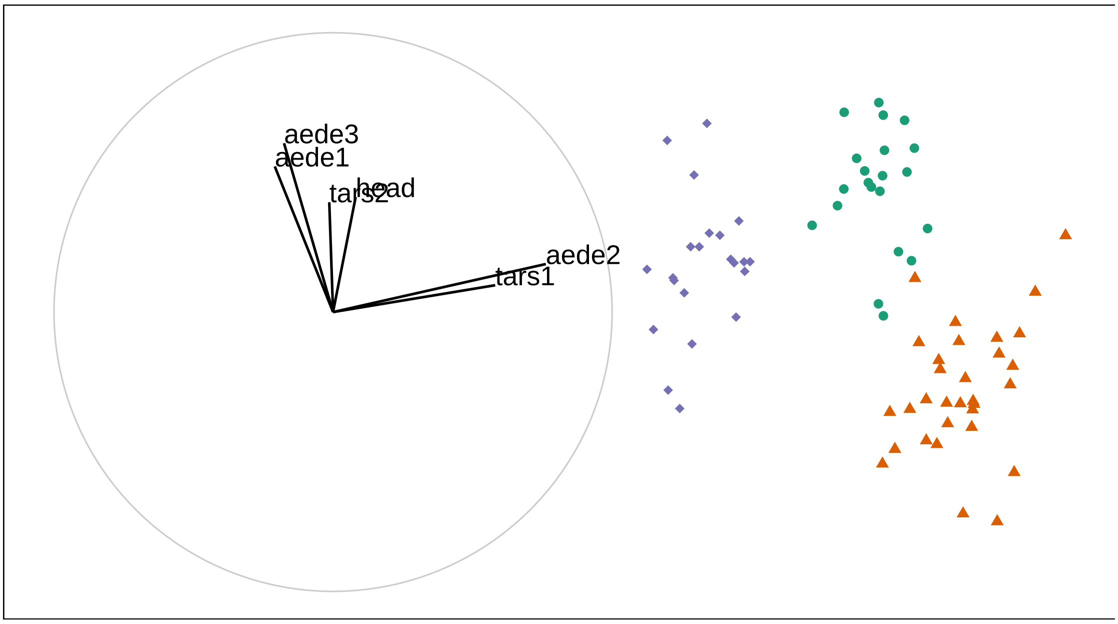


Figure 2.1: *Basis reference frame (left) and projected data (right) of standardized flea data. Basis identified by holes-index guided tour. The variables 'aede2' and 'tars1' contribute mostly in the x direction, whereas the other variables contribute mostly in the y direction. We'll select 'aede2' as our manipulation variable to see how the structure of the projection changes as we rotate 'aede2' into and out of the projection.*

of this data. The left plot displays the projection basis, while the right plot shows the projected data. The display of the basis has a unit circle with lines showing the horizontal and vertical contributions of each variable in the projection. Here is primarily tars1 and aede2 contrasting the other four variables. In the projected data there are three clusters, which have been colored, although not used in the optimization. The question that will be explored in the explanation of the algorithm is how important is aede2 to the separation of the clusters.

The left frame of figure 2.1 shows the reference frame for the basis. It describes the X and Y contributions of the basis as it projects from the 6 variable dimensions down to 2. Call `view_basis()` on a basis to produce a similar image as a `ggplot2` object. The right side shows how the data looks projected through this basis. You can project a single basis at any time through the matrix multiplication $\mathbf{X}_{[n,\ p]} * \mathbf{B}_{[p,\ d]} = \mathbf{P}_{d[n,\ d]}$ to such effect.

2.3.3 Step 1 Choose variable of interest

Select a manipulation variable, k. Initialize a zero vector e and set the k-th element set to 1.

$$\mathbf{e}_{[p,\ 1]} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

In figure 2.1, above, notice that the variables `tars1` and `aede2` are almost orthogonal to the other 4 variables and control almost all of the variation in the x axis of the projection. `Aede2` has a larger contribution in this basis, so we'll select it as the manip variable.

2.3.4 Step 2 Create the manip space

Use the Gram-Schmidt process to orthonormalize the concatenation of the basis and e yielding the manipulation space.

$$\begin{aligned} \mathbf{M}_{[p,\ d+1]} &= Orthonormalize_{GS}(\mathbf{B}_{[p,\ d]}|\mathbf{e}_{[p,\ 1]}) \\ &= Orthonormalize_{GS}\left(\begin{bmatrix} B_{1,\ 1} & \dots & B_{1,\ d} \\ B_{2,\ 1} & \dots & B_{2,\ d} \\ \vdots & \ddots & \vdots \\ B_{k,\ 1} & \dots & B_{k,\ d} \\ \vdots & \ddots & \vdots \\ B_{p,\ 1} & \dots & B_{p,\ d} \end{bmatrix} | \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}\right) \end{aligned}$$

In R it looks like the below chunk. `tourr::orthonormalise()` uses the Gram Schmidt process (rather than Householder reflection) to orthonormalize.

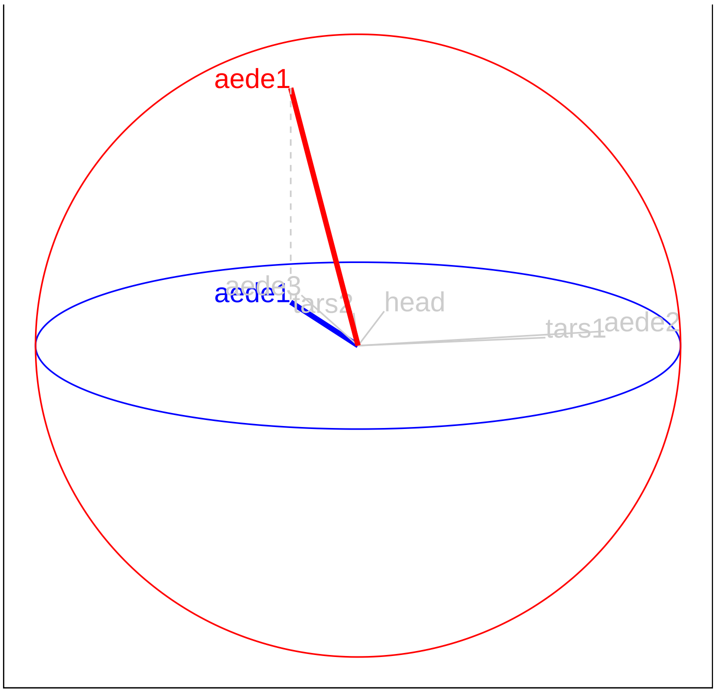


Figure 2.2: *Manipulation space for controlling the contribution of aede2 of standardized flea data. Basis was identified by holes-index guided tour. The out of plane axis, in red, shows how the manipulation variable can be rotated, while other dimensions stay embedded within the basis plane.*

```
e             <- rep(0, len = nrow(basis))
e[manip_var] <- 1
manip_space  <- tourr::orthonormalise(cbind(basis, e))
```

Adding an extra dimension to our basis plane allows for the manipulation of the specified variable while the others are kept fully within the basis plane. Orthonormalizing rescales the matrix without bringing the other variables into this new axis. An illustration of such can been seen below in figure 2.2.

Imagine being able to grab hold of the red axis and rotate it changing the projection onto the basis plane. This is what happens in a manual tour. By controlling the angle between the axis and the basis plane we change the contribution of the manipulation variable on the projection.

2.3.5 Step 3 Generate rotation

Define a set of values for ϕ_i, the angle of out of plane rotation, orthogonal to the projection plane. This corresponds to the angle between the red manipulation axis and the blue plane in figure 2.2.

For i in 1 to n_slides:

For each ϕ_i, post multiply the manipulation space by a rotation matrix, producing as many basis-projections.

$$\mathbf{P}_{b[p,\ d+1,\ i]} = \mathbf{M}_{[p,\ d+1]} \ * \ \mathbf{R}_{[d+1,\ d+1]}$$

For the $d = 2$ case:

$$= \begin{bmatrix} M_{1,\ 1} & \dots & M_{1,\ d} & M_{1,\ d+1} \\ M_{2,\ 1} & \dots & M_{2,\ d} & M_{2,\ d+1} \\ \vdots & \ddots & \vdots & \\ M_{p,\ 1} & \dots & M_{p,\ d} & M_{p,\ d+1} \end{bmatrix}_{[p,\ d+1]} \quad * \quad \begin{bmatrix} c_\theta^2 c_\phi s_\theta^2 & -c_\theta s_\theta (1 - c_\phi) & -c_\theta s_\phi \\ -c_\theta s_\theta (1 - c_\phi) & s_\theta^2 c_\phi + c_\theta^2 & -s_\theta s_\phi \\ c_\theta s_\phi & s_\theta s_\phi & c_\phi \end{bmatrix}_{[3,\ 3]}$$

Where:

θ is the angle that lies on the projection plane (*ie*. on the xy plane)

ϕ is the angle orthogonal to the projection plane (*ie*. in the z, direction)

c_θ is the cosine of θ

c_ϕ is the cosine of ϕ

s_θ is the sine of θ

s_ϕ is the sine of ϕ

In application: compile the sequence of ϕ_i and create an array (or long table) for each rotated manipulation space. ϕ is the angle relative to the ϕ_1, we find the transformation ϕ_i - ϕ_1 useful to think about ϕ relative to the basis plane. If the manip variable doesn't move as expected this is the first place to check.

```
for (phi in seq(seq_start, seq_end, phi_inc_sign)) {
  slide <- slide + 1
  tour[,, slide] <- rotate_manip_space(manip_space, theta, phi)[, 1:2]
}
```

In figure 2.3 we illustrate the sequence with 15 projected bases and highlight the manip variable on top, while showing the corresponding projected data points on the bottom. A dynamic version of this tour can be viewed online at https://nspyrison.netlify.com/thesis/flea_manualtour_mvar4/, will take a moment to load. This format of this figure and linking to dynamic version will be used again in section 2.5.

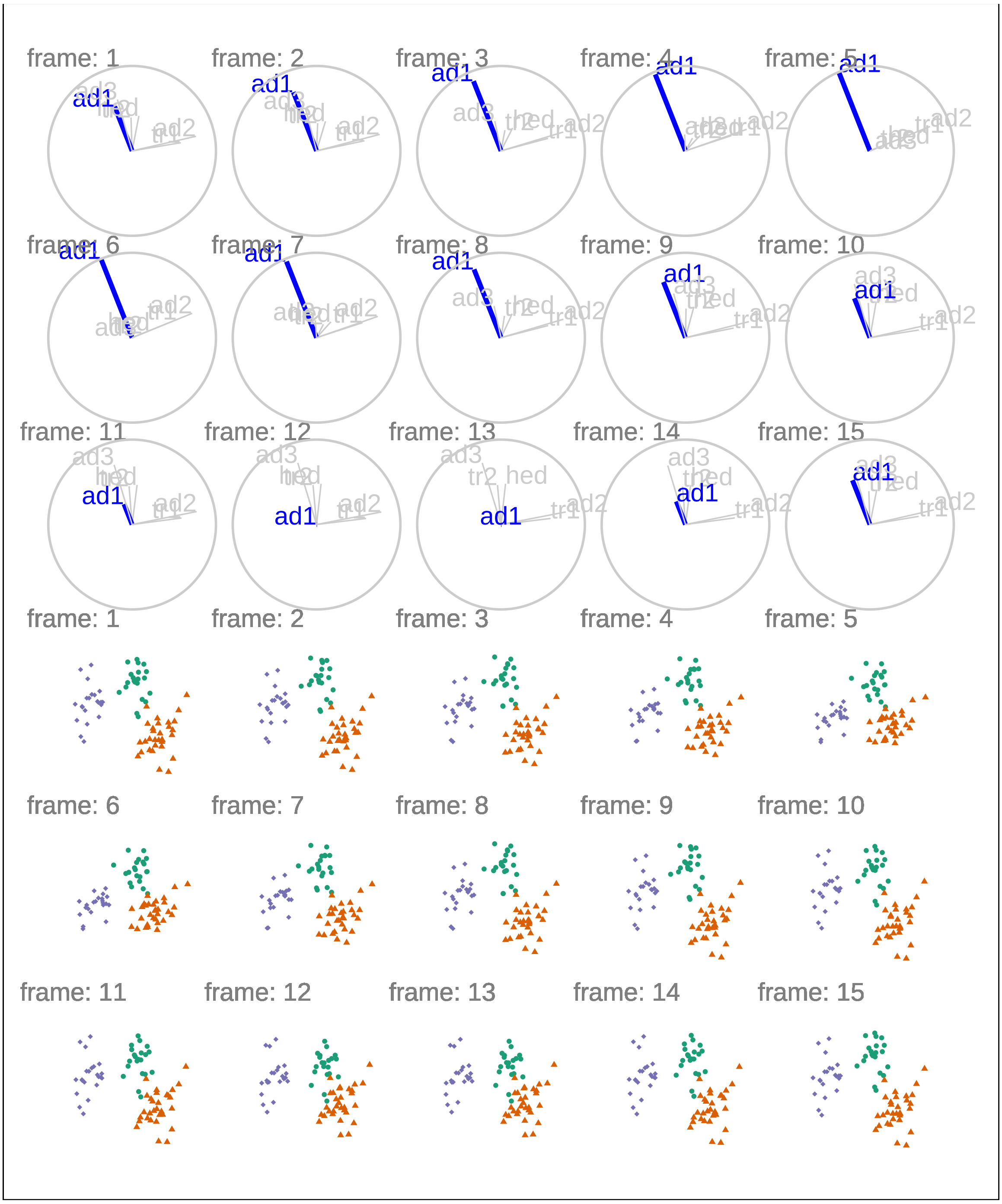


Figure 2.3: *Rotated manipulation spaces, a radial manual tour manipulating aded2 of standardized flea data. The manipulation variable, aede2, extends from its initial contribution to a full contribution to the projection before decreasing to zero, and then returning to its initial state. A dynamic version can be viewed at* `https://nspyrison.netlify.com/thesis/flea_manualtour_mvar4/`.

2.4 Display projection sequence

To get back to data-space pre-multiply each projection basis by the data for the projection in data-space.

$$\mathbf{P}_{d[n,\ d+1]} = \mathbf{X}_{[n,\ p]} \ * \ \mathbf{P}_{b[p,\ d+1]} \tag{2.1}$$

$$= \begin{bmatrix} X_{1,\ 1} & \dots & X_{1,\ p} \\ X_{2,\ 1} & \dots & X_{2,\ p} \\ \vdots & \vdots & \vdots \\ X_{n,\ 1} & \dots & X_{n,\ p} \end{bmatrix}_{[n,\ p]} * \begin{bmatrix} P_{b:1,\ 1} & P_{b:1,\ 2} & P_{b:1,\ 3} \\ P_{b:2,\ 1} & P_{b:2,\ 2} & P_{b:2,\ 3} \\ \vdots & \vdots & \vdots \\ P_{b:p,\ 1} & P_{b:p,\ 2} & P_{b:p,\ 3} \end{bmatrix}_{b[p,\ d+1]} \tag{2.2}$$

Plot the first 2 variables from each projection in sequence for an XY scatterplot. The remaining variable is sometimes linked to a data point aesthetic to produce depth cues used in conjunction with the XY scatterplot.

tourr utilizes R's base graphics for the display of tours. Use `render_plotly()` to display as an dynamic `plotly` Sievert (2018) object or `render_gganimate()` for a `gganimate` Pedersen and Robinson (2019) graphic. A third notable animation related package is `animation` Xie et al. (2018). It's not yet implemented in `spinifex` as it uses base graphics, whereas the former two are compatible with `ggplot2`.

Interaction with graphics in R is limited. Traditionally, all commands are passed to the R via calls to the console, conflicting with user engagement. Some recent packages have made advancement into this direction such as with the use of the R package `shinny`, which custom-made applications can be hosted either locally or remotely and interact with the R console, allowing for developers to code dynamic content interaction. To a lesser extent `plotly` offers static interactions with contained object, such as tool tips, brushing, and linking without communicating back to the R console.

Storing each data point and all of the overhead though goes into dynamic graphics if very inefficient. In the same way that we performed math the bases, that is the same approach storage and sharing tours. Consider the manual tour, we can store the salient features in 3 bases, where ϕ is at its starting, minimum, and maximum values. The frames in between can be interpolated by supplying angular speed or number of desired frames. By using the `tourr::save_history()` we can do just that. Save such tour path history and a single set of the data offers a performant storage and transferring.

2.5 Application

In a recent paper, Wang et al. (2018), the authors aggregate and visualize the sensitivity of hadronic experiments. The authors introduce a new tool, PDFSense, to aid in the visualization of parton distribution functions (PDF). The parameter-space of these experiments lies in 56 dimensions, $\delta \in \mathbb{R}^{56}$, and are presented in this work in 3D subspaces of the 10 first principal components and non-linear embeddings.

The work in Cook, Laa, and Valencia (2018) applies touring for discern finer structure of this sensitivity. Table 1 of Cook et. al. summaries the key findings of PDFSense & TFEP (TensorFlow embedded projection) and those from touring. The authors selected the 6 first principal components, containing 48% of the variation held within the full data when centered, but not sphered. This data contained 3 clusters: jet, DIS, and VBP. Below pick up from the projections used in their figures 7 and 8 (jet and DIS clusters respectively) and apply manual tours to explore the local structure with finer precision.

2.5.1 Jet cluster

The jet cluster is of particular interest as it contains the largest data sets and is found to be important in Wang et al. (2018). The jet cluster resides in a smaller dimensionality than the full set of experiments with 4 principal components explaining 95% of its variation (Cook, Laa, and Valencia (2018)). We subset the data down to ATLAS7old and ATLAS7new to narrow in on 2 groups with a reasonable number of observations and occupy different parts of the subspace. Below, we perform radial manual tours on various principal components within this scope. In PC3 and PC4 are manipulated in figure 2.4 and figure 2.5 respectively. Manipulating PC3, where varying the angle of rotation brings interesting features in-to and out of the center mass of the data, is interesting than the manipulation of PC4, where features are mostly independent of the manip var.

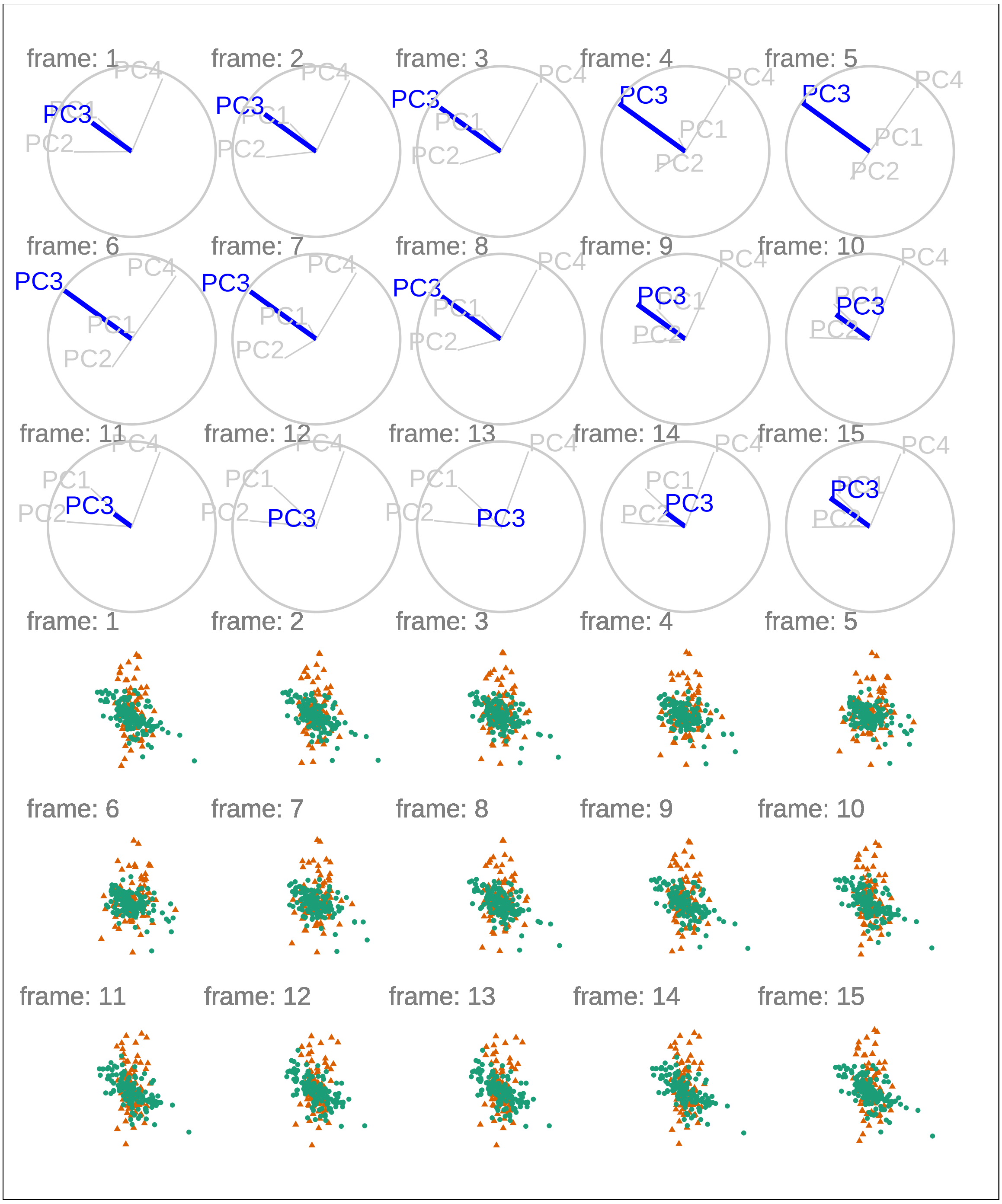


Figure 2.4: *Jet cluster, radial manual tour of PC3. Colored by experiment type: 'ATLAS7new' in green and 'ATLAS7old' in orange. When PC3 fully contributes to the projection ATLAS7new (green) occupies unique space and several outliers are identifiable. Zeroing the contribution from PC3 to the projection hides the outliers and indeed all observations with ATLAS7new are contained within ATLAS7old (orange). A dynamic version can be viewed at* `https://nspyrison.netlify.com/thesis/jetcluster_manualtour_pc3/`.

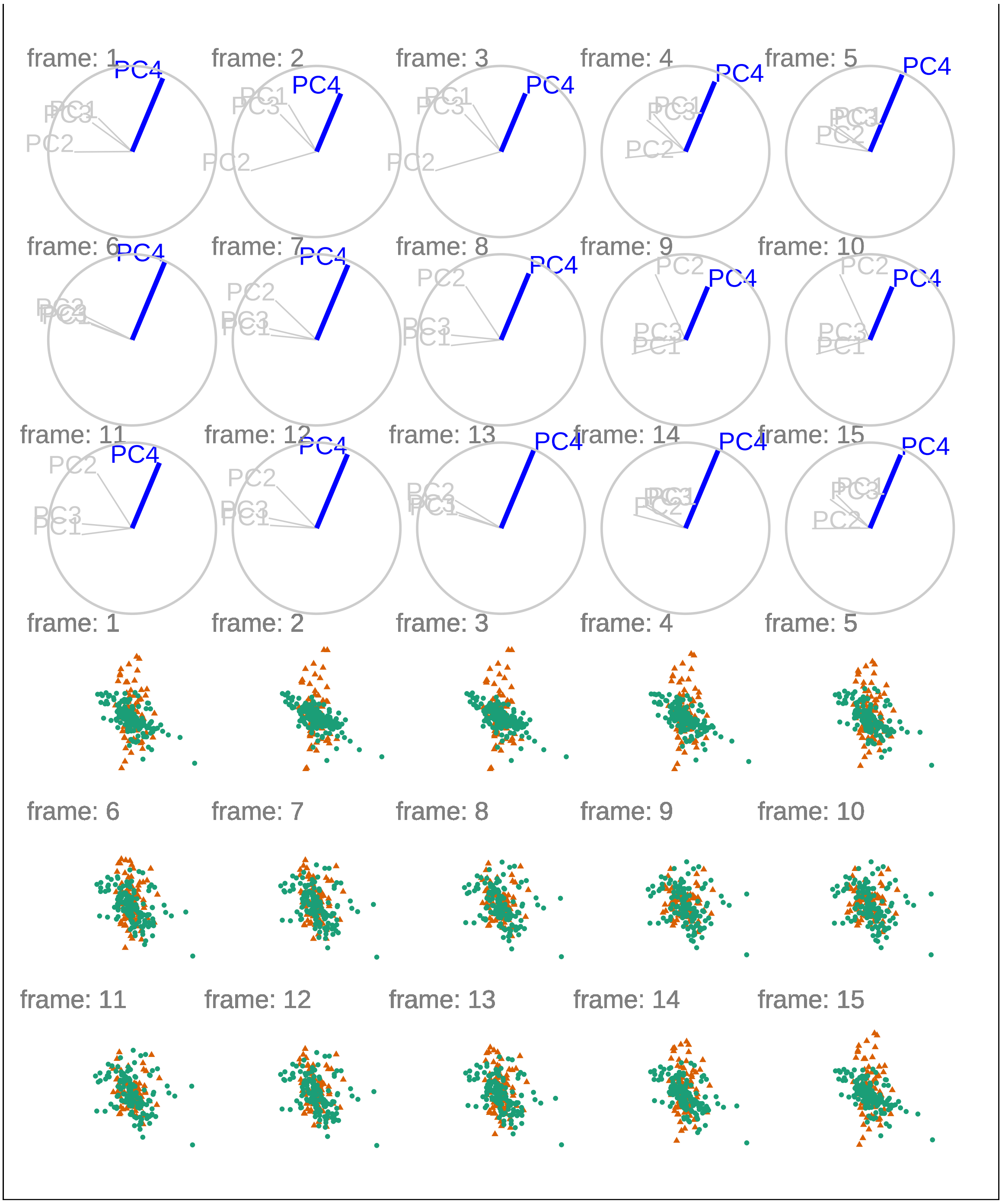


Figure 2.5: *Jet cluster, radial manual tour of PC4. Colored by experiment type: 'ATLAS7new' in green and 'ATLAS7old' in orange. This tour contains less interesting information ATLAS7new (green) has points that are right and left of ATLAS7old, while most points occupy the same projection space, regardless of the contribution of PC4. A dynamic version can be viewed at* `https://nspyrison.netlify.com/thesis/jetcluster_manualtour_pc3/`.

Jet cluster manual tours manipulating each of the principal components can be viewed from the links: PC1, PC2, PC3, and PC4.

2.5.2 DIS cluster

We perform a manual tour on this data, manipulating PC6 as depicted in figure 2.6. Looking at several frames we see that DIS HERA lie mostly on a plane. When PC6 has full contributions, we see the dimuon SIDIS in purple is almost orthogonal to the DIS HERA (green). Yet the contribution of PC6 is zeroed the dimuon SIDIS data occupy the same space as the DIS HERA data. A dynamic version of this manual tour can be found at: `https://nspyrison.netlify.com/thesis/discluster_manualtour_pc6/`. The page takes a bit to load, as the animation is several megabytes.

This is different story than if we had selected a different variable to manipulate. In figure 2.7 we manipulate PC2.

DIS cluster manual tours manipulating each of the principal components can be viewed from the links: PC1, PC2, PC3, PC4, PC5, and PC6.

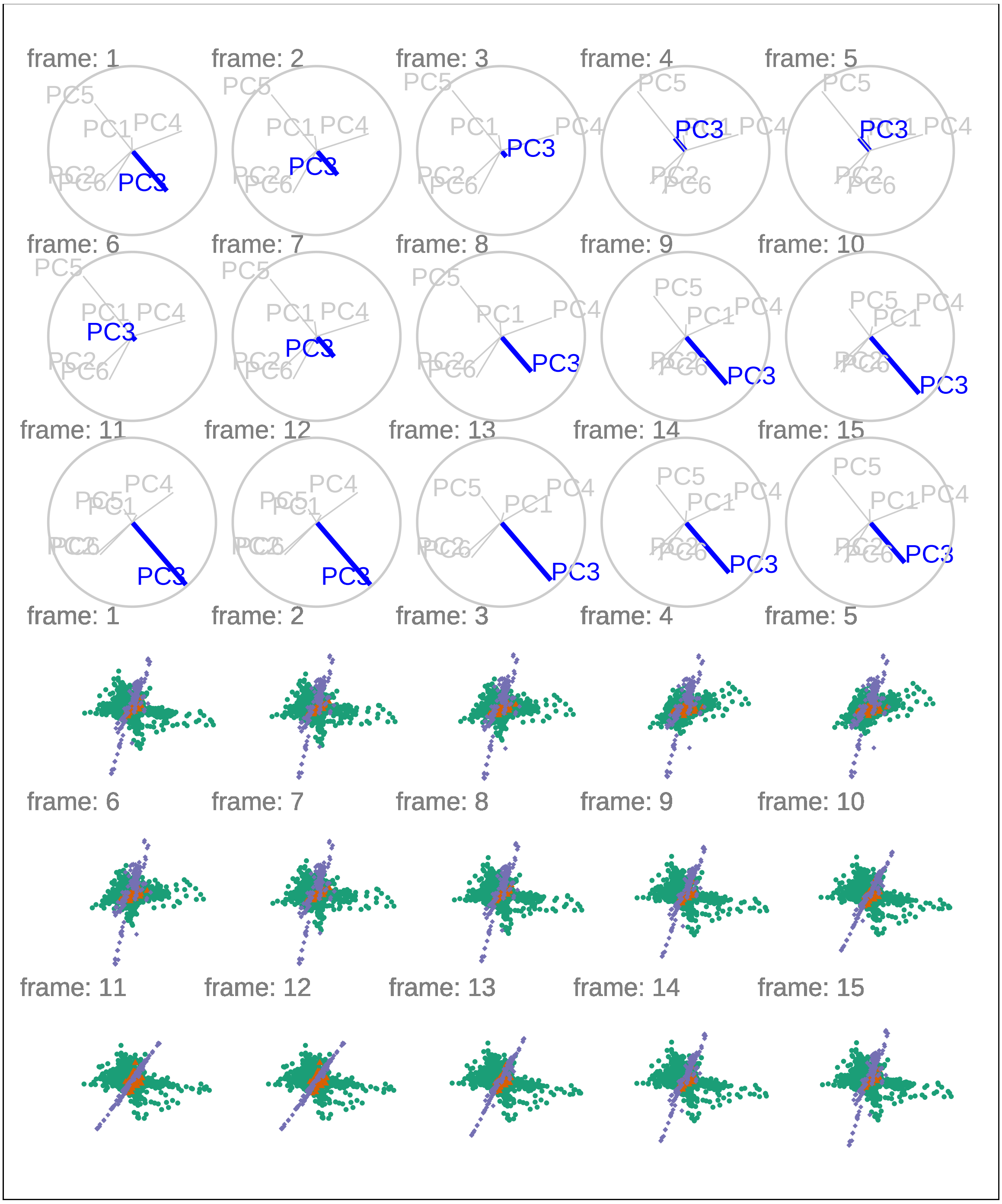


Figure 2.6: *DIS cluster, radial manual tour of PC6. colored by experiment type: 'DIS HERA1+2' in green, 'dimuon SIDIS' in purple, and 'charm SIDIS' in orange. When the contribution PC 6 is large we see that dimuon SIDIS (purple) data are nearly orthogonal to DIS HERA (green) data. As the data is rotated, we can also see that DIS HERA (green) practically lie on a plane in this 6-d subspace. When the contribution of PC6 is near zero, dimonSIDIS (purple) occupies the same space as the DIS HERA data. A dynamic version can be viewed at* `https://nspyrison.netlify.com/thesis/discluster_manualtour_pc6/`.

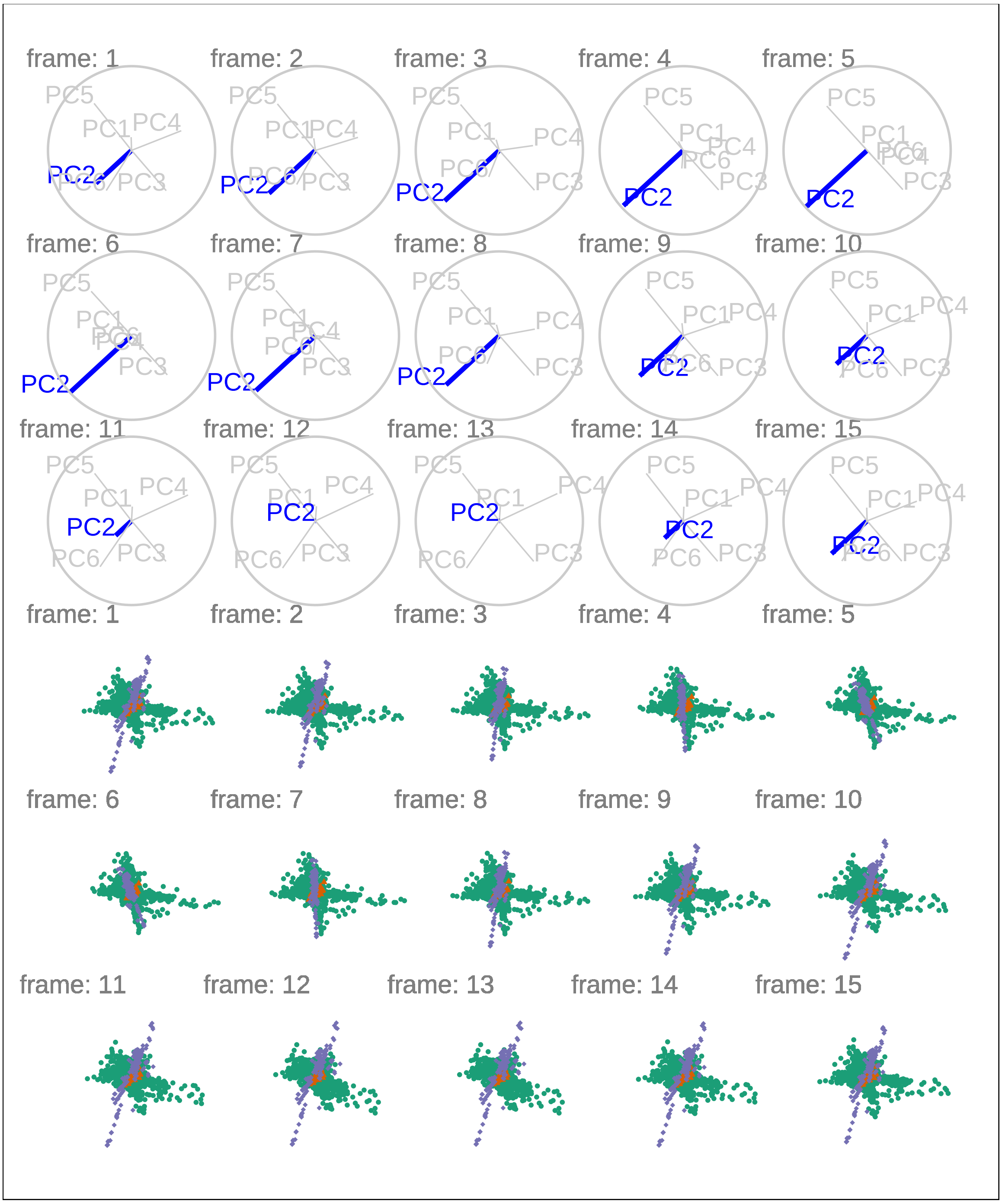


Figure 2.7: *DIS cluster, radial manual tour of PC2. Colored by experiment type: 'DIS HERA1+2' in green, 'dimuon SIDIS' in purple, and 'charm SIDIS' in orange. The structure of previously described plane of DIS HERA (green) and nearly orthogonal dimuon SIDIS (purple) is present, however the manipulating PC2 does not give a head-on view of either, a less useful manual tour than that of PC6. A dynamic version can be viewed at* `https://nspyrison.netlify.com/thesis/discluster_manualtour_pc2/`.

2.6 Source code and usage

This article was created `bookdown` (Xie (2016)) using `rmarkdown` (Xie, Allaire, and Grolemund (2018)), with code generating the examples inline, and the source files can be found at github.com/nspyrison/confirmation/.

The source code for the `spinifex` package can be found at github.com/nspyrison/spinifex/. To install the package in R, run:

```
# install.package("devtools")
devtools::install_github("nspyrison/spinifex")
```

2.7 Discussion

This work has described an algorithm and package for exploring conducting a manual tour, from a 2D projection, to explore the sensitivity of structure to the contributions of a variable.

Future work on the algorithm and package would include developing it to work with arbitrary projection dimension, enabling the method to operate on other displays like parallel coordinates, and implementing the unconstrained manual control, called oblique in Cook and Buja (1997).

The Givens rotations and Householder reflections as outlined in Buja et al. (2005) may provide a way to conduct higher dimensional manual control. In a Givens rotation, the x and y components ($ie.\theta = 0,\ pi/2$) of the in-plane rotation are calculated separately and would be applied sequentially to produce the radial rotation. Householder reflections define reflection axes to project points on to the axes and generate rotations.

The *tourr* package provides several d-dimensional graphic displays including Andrews curves, Chernoff faces, parallel coordinate plots, scatterplot matrix, and radial glyphs. Having manual controls available for these types of displays would require a dimensionally-generalized rotation matrix.

Development of a graphical user interface, e.g. *shiny* app, would make the *spinifex* package more flexible. The user could easily switch between variables to control, adjust the step size to make smoother rotation sequences, or save any state to continue to continue to explore the contributions of other variables.

Chapter 3

What benefit does UCS have over popular alternatives?

High dimensional data and models are ubiquitous but viewing them in data space is not trivial. Currently it is common practice to use Principal Component Analysis (PCA), Multi-Dimensional Scaling (MDS), or a non-linear embedding like t-distributed neighbor embeddings (tSNE). These methods are all unsupervised projections of multivariate spaces to a single, static lower embedding. PCA is a linear transformation that orients linear combinations of the variables into basis components and orders them according to amount of variation shown. MDS has linear and non-linear versions that compare the distances between pairwise variables. tSNE is a nonlinear technique that iterates epochs of: 1) constructing a probability distributions for selecting neighboring data and 2) minimizing Kullback-Leibler divergence (a measure of relative entropy).

Unfortunately, static linear projections necessarily cut variation in the components not shown, while non-linear techniques lose transparency back to the original variable-space. Touring preserves transparency to variable-space and keeps variation in tack. By providing user-controlled steering of tour we should be able to provide finer structural exploration than the alternatives.

In this future work I plan to perform a case study between UCS leading alternatives. Design space includes data sets, techniques, and measures of comparison.

Chapter 4

How can we extend UCS to 3D?

Touring involves linear embeddings in typically $d = 2$ dimensions. While Nelson, Cook, and Cruz-Neira (1998) and Arms, Cook, and Cruz-Neira (1999) have explored the efficacy of 2D vs 3D display, they have done so on 2D embeddings, I will generalize the UCS to scalable dimensions in R, built upon the existing packages *tourr* and *spinifex*. This will allow for the backend work of 3D rotations.

Recently Cordeil et al. (2017) shared immersive data analysis written in C# to be used in conjunction with the 3D compatible game engine Unity. The outcome of this work was refined and generalized in the Immersive Analytics Toolkit (IATK), Cordeil (2019).

This future work in an exploratory design extending UCS to 3D projections and integrating the calling R scripts with the existing IATK for a compatible front end to be used across display devices.

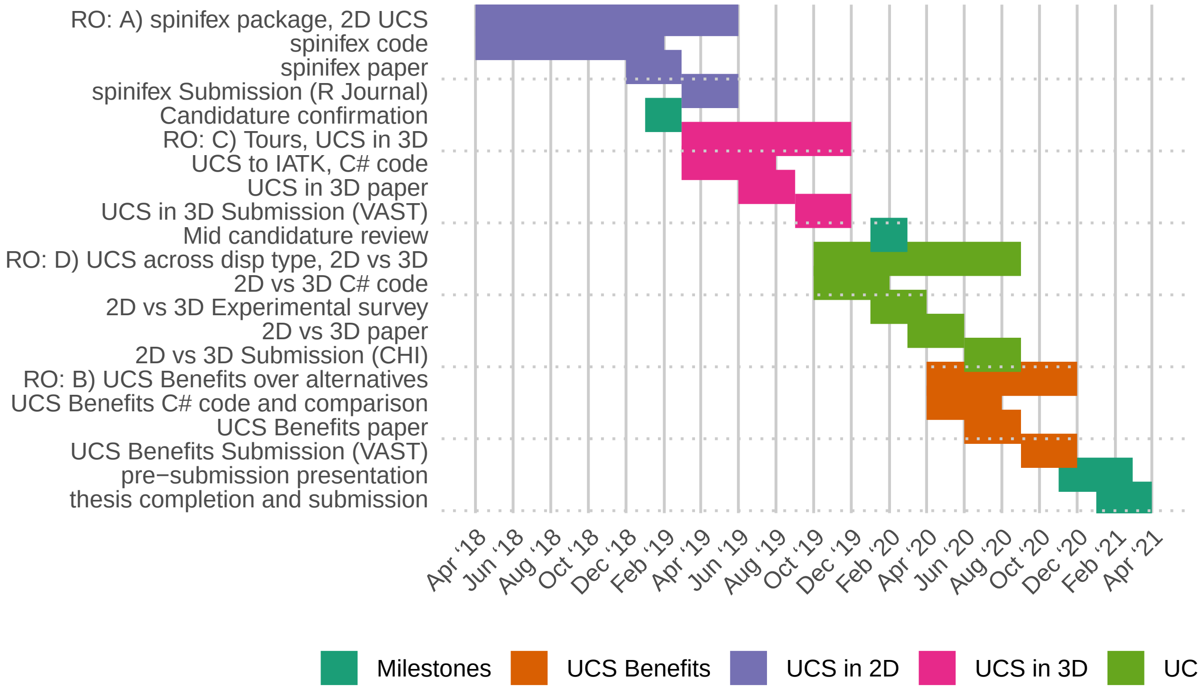
Chapter 5

Does 3D UCS provide benefits over UCS in 2D?

The bulk of past touring endeavors have existed whole in 2D, with the exceptions of Nelson, Cook, and Cruz-Neira (1998) and Arms, Cook, and Cruz-Neira (1999) whom performed a small ($n = 15$) experimental study comparing tasks performed across 2D and 3D touring displays. The XGobi interface was used on a standard 2D monitor while VRGobi (on the C2 setup) was used with head-tracked binocular VR. The 3 accuracy tasks: clustering, intrinsic data dimensionality, and radial sparseness were recorded along with the speed of a brushing data. Accuracy was the same for the dimensionality task, while the d3 display out performed 2D on clustering, and even more so on the radial sparsity. However, time taken to brush a cluster was less than half the time in 2D display as compared with 3D.

The results of Wagner Filho et al. (2018), Nelson, Cook, and Cruz-Neira (1998) and, Arms, Cook, and Cruz-Neira (1999) cast positive light on 3D and immerse spaces improving perception of project high dim data. After implementing touring and UCS in 3D spaces (**RO C**), I plan to explore the efficacy of doing so with the following empirical study: comparison of 3D touring across display dimension in 4 instances: standard 2D monitor, stereoscopic 3D monitor (on a zSpace 200), and head-mounted VR goggle (HTC VIVE), and immersion in a CAVE environment. Implementation in the game engine Unity will allow for a standardized user interface. Tasks of structure perception will be conducted across 2 data sets of high energy physics data already in publication (Wang et al., 2018; Cook, Laa, and Valencia, 2018) and discussed in chapter 2. Task order will be randomly assigned to minimize learning bias. Participants will perform all tasks on each display devices, for each of the data sets. Time, and accuracy will be tracked, and participants will be asked to fill out a small survey with demographic data and subjective experience on a 5-point Likert scale. The design space of this study includes display type, task type, survey questions, familiarity with 3D, familiarity with linear projections.

PhD timeline



Note RO #B logically would fit before RO #C & D but is almost a foregone conclusion based on the methodology's contained variation and transparency to original variable-space. I move this research to the end, such that this study would be tight on time if such a case arises.

Also submitted via online milestone form:

- FIT 5144 hours
  - >120 hours **Tracked, documentation needs cleanup**, due at mid-candidature review
- WES Academic record

  - FIT6021: 2018 S2, **Completed** with Distinction
  - FIT5144: 2019 S1+2, **Upcoming**, due at mid-candidature review
  - FIT5113: 2018 S2, **Exemption submitted**

- myDevelopment - IT: Monash Doctoral Program - Compulsory Module
  - Monash Graduate Research Student Induction: **Completed**
  - Research Integrity - Choose the Option most relevant: **Completed** (2 required of 4)
  - Faculty Induction: **Content unavailable** (9/02/2019: "Currently being updated and will be visible in this section soon".)

Acknowledgements

This thesis report is generated using R(R Core Team, 2018), bookdown [xie_bookdown:_2016] and rmarkdown (Xie, Allaire, and Grolemund, 2018), where code and explanation are kept together, following reproducible research practice. The code producing this document can be found at: `https://github.com/nspyrison/Confirmation`.

Bibliography

Andrews, DF (1972). Plots of High-Dimensional Data. *Biometrics* **28**(1), 125–136. (Visited on 12/19/2018).

Anscombe, FJ (1973). Graphs in Statistical Analysis. *The American Statistician* **27**(1), 17–21. (Visited on 12/19/2018).

Arms, L, D Cook, and C Cruz-Neira (1999). The benefits of statistical visualization in an immersive environment. In: *Virtual Reality, 1999. Proceedings., IEEE*. IEEE, pp.88–95.

Asimov, D (1985). The grand tour: a tool for viewing multidimensional data. *SIAM journal on scientific and statistical computing* **6**(1), 128–143.

Becker, RA and WS Cleveland (1987). Brushing Scatterplots. *Technometrics* **29**(2), 127–142. (Visited on 01/10/2019).

Buja, A, D Cook, D Asimov, and C Hurley (2005). "Computational Methods for High-Dimensional Rotations in Data Visualization". en. In: *Handbook of Statistics*. Vol. 24. Elsevier, pp.391–413. http://linkinghub.elsevier.com/retrieve/pii/S0169716104240147 (visited on 04/15/2018).

Buja, A, C Hurley, and JA McDonald (1987). A data viewer for multivariate data. In: *Colorado State Univ, Computer Science and Statistics. Proceedings of the 18 th Symposium on the Interface p 171-174(SEE N 89-13901 05-60)*.

Carr, DB and WL Nicholson (1988). 'Explor4: A Program for Exploring Four-Dimensional Data Using Stereo-Ray Glyphs, dimensional constraints, rotation, and masking. *Cleveland and McGill (1988)*, 309–329.

Carr, D, E Wegman, and Q Luo (1996). ExplorN: Design considerations past and present. **129**.

Carreira-Perpinán, MA (1997). A review of dimension reduction techniques. *Department of Computer Science. University of Sheffield. Tech. Rep. CS-96-09* **9**, 1–69.

Chernoff, H (1973). The Use of Faces to Represent Points in K-Dimensional Space Graphically. *Journal of the American Statistical Association* **68**(342), 361–368. (Visited on 01/05/2019).

Cook, D and A Buja (1997). Manual Controls for High-Dimensional Data Projections. *Journal of Computational and Graphical Statistics* **6**(4), 464–480. (Visited on 04/15/2018).

Cook, D, A Buja, and J Cabrera (1993). Projection Pursuit Indexes Based on Orthonormal Function Expansions. *Journal of Computational and Graphical Statistics* **2**(3), 225–250. (Visited on 01/07/2019).

Cook, D, A Buja, J Cabrera, and C Hurley (1995). Grand Tour and Projection Pursuit. en. *Journal of Computational and Graphical Statistics* **4**(3), 155. (Visited on 05/27/2018).

Cook, D, U Laa, and G Valencia (2018). Dynamical projections for the visualization of PDFSense data. *Eur. Phys. J. C* **78**(9), 742.

Cook, D, DF Swayne, and A Buja (2007). *Interactive and Dynamic Graphics for Data Analysis: With R and GGobi*. en. Google-Books-ID: 34DL7lR_4CoC. Springer Science & Business Media.

Cordeil, M (2019). *Immersive Analytics Toolkit*. original-date: 2017-02-16T05:25:32Z. https://github.com/MaximeCordeil/IATK (visited on 02/04/2019).

Cordeil, M, A Cunningham, T Dwyer, BH Thomas, and K Marriott (2017). ImAxes: Immersive Axes As Embodied Affordances for Interactive Multivariate Data Visualisation. In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. UIST '17. New York, NY, USA: ACM, pp.71–83. http://doi.acm.org/10.1145/3126594.3126613 (visited on 08/20/2018).

Fisherkeller, MA, JH Friedman, and JW Tukey (1974). PRIM-9: An Interactive Multidimensional Data Display and Analysis System.

Friedman, J and J Tukey (1974). A Projection Pursuit Algorithm for Exploratory Data Analysis. en. *IEEE Transactions on Computers* **C-23**(9), 881–890. (Visited on 06/22/2018).

Gracia, A, S González, V Robles, E Menasalvas, and T Von Landesberger (2016). New insights into the suitability of the third dimension for visualizing multivariate/multidimensional data: A study based on loss of quality quantification. *Information Visualization* **15**(1), 3–30.

Grimm, K (2017). *mbgraphic: Measure Based Graphic Selection*. https://CRAN.R-project.org/package=mbgraphic (visited on 02/07/2019).

Grinstein, G, M Trutschl, and U Cvek (2002). High-Dimensional Visualizations. en, 14.

Heer, J, M Bostock, and V Ogievetsky (2010). A tour through the visualization zoo. en. *Communications of the ACM* **53**(6), 59. (Visited on 05/31/2018).

Huang, B, D Cook, and H Wickham (2012). tourrGui: A gWidgets GUI for the tour to explore high-dimensional data using low-dimensional projections. *Journal of Statistical Software* **49**(6), 1–12.

Huber, PJ (1985). Projection Pursuit. en. *The Annals of Statistics* **13**(2), 435–475.

Huh, MY and K Song (2002). DAVIS: A Java-based Data Visualization System. en. *Computational Statistics* **17**(3), 411–423. (Visited on 01/06/2019).

Hurley, C and A Buja (1990). Analyzing High-Dimensional Data with Motion Graphics. *SIAM Journal on Scientific and Statistical Computing* **11**(6), 1193–1211. (Visited on 11/27/2018).

Laa, U and D Cook (2019). Using tours to visually investigate properties of new projection pursuit indexes with application to problems in physics. *arXiv:1902.00181 [physics, stat]*. arXiv: 1902.00181. (Visited on 02/04/2019).

Lee, EK and D Cook (2010). A projection pursuit index for large p small n data. en. *Statistics and Computing* **20**(3), 381–392. (Visited on 02/13/2019).

Lee, EK, D Cook, S Klinke, and T Lumley (2005). Projection Pursuit for Exploratory Supervised Classification. *Journal of Computational and Graphical Statistics* **14**(4), 831–846. (Visited on 01/07/2019).

Lee, JM, J MacLachlan, and WA Wallace (1986). The effects of 3D imagery on managerial data interpretation. *MIS Quarterly*, 257–269.

Lubischew, AA (1962). On the use of discriminant functions in taxonomy. *Biometrics*, 455–477.

Marriott, K, F Schreiber, T Dwyer, K Klein, NH Riche, T Itoh, W Stuerzlinger, and BH Thomas (2018). *Immersive Analytics*. en. Google-Books-ID: vaVyDwAAQBAJ. Springer.

Matejka, J and G Fitzmaurice (2017). Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing. en. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. Denver, Colorado, USA: ACM Press, pp.1290–1294. http://dl.acm.org/citation.cfm?doid=3025453.3025912 (visited on 12/19/2018).

McDonald, JA (1982). INTERACTIVE GRAPHICS FOR DATA ANALYSIS.

Munzner, T (2014). *Visualization analysis and design*. AK Peters/CRC Press.

Nelson, L, D Cook, and C Cruz-Neira (1998). XGobi vs the C2: Results of an Experiment Comparing Data Visualization in a 3-D Immer- sive Virtual Reality Environment with a 2-D Workstation Display. en. *Computational Statistics* **14**(1), 39–52.

Ocagne, Md (1885). *Coordonnées parallèles et axiales. Méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèles, par Maurice d'Ocagne, ...* French. OCLC: 458953092. Paris: Gauthier-Villars.

Pedersen, TL and D Robinson (2019). *gganimate: A Grammar of Animated Graphics*. http://github.com/thomasp85/gganimate.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Scott, DW (1985). Averaged shifted histograms: effective nonparametric density estimators in several dimensions. *The Annals of Statistics*, 1024–1040.

Scott, DW (1995). Incorporating density estimation into other exploratory tools. In: *ASA Proceedings of the Section on Statistical Graphics', American Statistical Association, Alexandria, VA*. Citeseer, pp.28–35.

Sedlmair, M, T Munzner, and M Tory (2013). Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Transactions on Visualization & Computer Graphics* (12), 2634–2643.

Siegel, JH, EJ Farrell, RM Goldwyn, and HP Friedman (1972). The surgical implications of physiologic patterns in myocardial infarction shock. English. *Surgery* **72**(1), 126–141. (Visited on 01/05/2019).

Sievert, C (2018). *plotly for R*. https://plotly-book.cpsievert.me.

Sutherland, P, A Rossini, T Lumley, N Lewin-Koh, J Dickerson, Z Cox, and D Cook (2000). Orca: A Visualization Toolkit for High-Dimensional Data. *Journal of Computational and Graphical Statistics* **9**(3), 509–529. (Visited on 01/10/2019).

Swayne, DF, D Cook, and A Buja (1991). *Xgobi: Interactive Dynamic Graphics In The X Window System With A Link To S*.

Swayne, DF, DT Lang, A Buja, and D Cook (2003). GGobi: evolving from XGobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*. Data Visualization **43**(4), 423–444. (Visited on 12/19/2018).

Tierney, L (1990). *LISP-STAT: An Object Oriented Environment for Statistical Computing and Dynamic Graphics*. eng. Wiley Series in Probability and Statistics. New York, NY, USA: Wiley-Interscience.

Tory, M, AE Kirkpatrick, MS Atkins, and T Moller (2006). Visualization task performance with 2D, 3D, and combination displays. *IEEE transactions on visualization and computer graphics* **12**(1), 2–13.

Wagner Filho, J, M Rey, C Freitas, and L Nedel (2018). Immersive Visualization of Abstract Information: An Evaluation on Dimensionally-Reduced Data Scatterplots. In:

Wang, BT, TJ Hobbs, S Doyle, J Gao, TJ Hou, PM Nadolsky, and FI Olness (2018). Visualizing the sensitivity of hadronic experiments to nucleon structure. *arXiv preprint arXiv:1803.02777*.

Ware, C (2000). Designing with a 2$1/2$D attitude. *Information Design Journal* **10**(3), 258–265.

Wegman, EJ (2003). Visual data mining. en. *Statistics in Medicine* **22**(9), 1383–1397. (Visited on 12/19/2018).

Wegman, E, W Poston, and J Solka (2001). *Pixel Tours*. University of Minnesota. `https://ima.umn.edu/2001-2002/W11.12-15.01/18492` (visited on 01/10/2019).

Wickens, CD, DH Merwin, and EL Lin (1994). Implications of graphics enhancements for the visualization of scientific data: Dimensional integrality, stereopsis, motion, and mesh. *Human Factors* **36**(1), 44–61.

Wickham, H, D Cook, and H Hofmann (2015). Visualizing statistical models: Removing the blindfold: Visualizing Statistical Models. en. *Statistical Analysis and Data Mining: The ASA Data Science Journal* **8**(4), 203–225. (Visited on 03/16/2018).

Wickham, H, D Cook, H Hofmann, and A Buja (2011). **tourr** : An *R* Package for Exploring Multivariate Data with Projections. en. *Journal of Statistical Software* **40**(2). (Visited on 11/23/2018).

Xie, Y (2016). *bookdown: Authoring Books and Technical Documents with R Markdown*. Boca Raton, Florida: Chapman and Hall/CRC. `https://github.com/rstudio/bookdown`.

Xie, Y, JJ Allaire, and G Grolemund (2018). *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman and Hall/CRC. `https://bookdown.org/yihui/rmarkdown`.

Xie, Y, C Mueller, L Yu, and W Zhu (2018). *animation: A Gallery of Animations in Statistics and Utilities to Create Animations*. `https://yihui.name/animation`.